Implementing Integrated Services of Networked Home Appliances Using Service Oriented Architecture

Presenter: Hiroshi Igaki
Software Engineering Lab.

HNS is a system consisting of several networked home appliances and adds more value to the daily life.

The appliances in HNS are controlled together to provide integrated services.
- DVD, TV, Lights, Air Conditioner, Thermometer, etc.

HNS Integrated Services

Auto-illumination Service

1. User turns on the Light
2. Light adjusts its brightness to optimal based on the current illumination.

DVD Theatre Service

1. User turns on the DVD
2. TV is turned on and Speaker Channel is set at DVD-mode
3. Light becomes dark, adjusted based on illuminometer

Conventional HNS (Server Centralized Architecture)

-Almost all the conventional HNS use the home server, in order to realize integrated service.

Problems on the conventional HNS: Interoperability(1/4)

- In order to operate appliance, the home server needs to send control command based on these protocols.

Proprietary protocol
Proprietary protocol
Proprietary protocol
Proprietary protocol
Proprietary protocol
Proprietary protocol

In order to operate appliance, the home server needs to send control command based on these protocols.

If all appliances use same protocol, implementation of the home server is so simple.

Gateway Mechanism

Home Server
DVD
TV
Speaker
Light
Lumino meter
Proprietary protocol
Proprietary protocol
Proprietary protocol
Proprietary protocol
Proprietary protocol
Proprietary protocol

Home Network System (HNS)

- DVDs, TV, Lights, Air Conditioner, Thermometer, etc.

Home Server
DVD
TV
Speaker
Light
Lumino meter
Proprietary protocol
Proprietary protocol
Proprietary protocol
Proprietary protocol
Proprietary protocol
Proprietary protocol

The home server sends control commands to the end devices to control all devices directly.
Problems on the conventional HNS: System Extention (2/4)

- In HNS, users may add or modify the contents of integrated services and appliance environments.
- Gateway Mechanism
  - Proprietary protocol
  - Proprietary protocol

Problems on the conventional HNS: Load Concentration (3/4)

- Every integrated service needs the Home Server. The number of appliances reflects heavy workload of the server.

Problems on the conventional HNS: Reliability (4/4)

- A crash of the server makes all the integrated services unavailable.

Research Goal

Goal
- Develop improved HNS
  - Interoperability, Extendibility, Load-balancing, Reliability

Approach
- Design the SOA-based integrated services with concrete service scenarios
- Evaluate the integrated services with a graph-based method

Example Service Scenarios

- SS1: Auto Illumination Service
  - The brightness of the light is automatically adjusted based on the current intensity of illumination with the illuminometer.
- SS2: Coming Home Service
  - If the user enters a room from the door, the light is turned on.
- SS3: DVD Theatre Service
  - When the user turns on the DVD player, the light becomes dark. Then, the TV and the speaker start in the DVD mode.
- SS4: Auto-TV Service
  - When the user watches the TV, the speaker is turned on.
- SS5: Ringing and Mute Service
  - While the user is watching the TV, if the telephone rings, then the volume of the speaker becomes small.
- SS6: Auto Temperature Control Service
- SS7: Coming Home Service about Temperature
- SS8: Sleep Service

Key Idea (1/2)

- Standard Communication and Loose Coupling
  - Service layer provides appliance features as a service (a set of exported methods)
Key Idea (2/2)

- Autonomous Orchestration without server
  - Service Layer enables direct and autonomous communications among appliances without any centralized servers.

User

- Illuminometer
  - Exported Methods
    - getIllumination
    - setIllumination
  - Device layer
    - proprietary protocols
    - device layer
    - generic protocols

Illuminometer

- Exported Methods
  - Light
    - ON
    - setIllumination
  - Device
    - User

- Service Layer
  - Exported Methods
    - Light
      - ON
      - setIllumination
      - getIllumination
  - Device
    - User

- Implementation of HNS prototype
  - Service Layer
    - Java Web Service
      - Web server: Jakarta Tomcat 4.1.18
      - Apache-AXIS 1.1
      - Java2 SDK SE 1.4.1_02
  - Device Layer
    - Virtual Device component
      - Implemented with java beans

Example of Sequence diagram

Light

- Exported Methods
  - ON
  - setIllumination

Illuminometer

- Exported Methods
  - getIllumination
  - setIllumination

- Context
  - SSID
  - ServiceURI
  - Pre/Post
  - methodName
  - paramName
  - paramType
  - paramValue

Example of Sequence diagram

Light Component Example

- Exported Method
  - setIllumination
  - ON

Illuminometer Service

- SMI
  - Definition
  - DMI

- Virtual Light Device

- SMI Definition of Light Service

Implementation Template

- Service Template
  - SMI Definition
  - preProcess
  - deviceMethod
  - postProcess

- DMI
  - is fixed

- is implemented as preProcess and postProcess

- Contents are determined by SMI Def

Implementation: Service template

- Exported Method
  - preProcess
  - deviceMethod
  - postProcess

- DMI
  - SMI

- Virtual Device

- SMI Definition

- SMIDV
**Design metrics of HNS**

- Reliability
- Workload
- Coupling

**n-reliability based on SDP Algorithm**

- Present each service scenario as a labeled directed graph
- For a given HNS with integrated service scenarios we define n-reliability as the probability that at least n service scenarios are operational in the HNS.
- Reliability of each node and edge are given, the SDP calculates reliability that at least one of specified set of service scenarios.

**n-Reliability: Result**

- n-reliability for SCA becomes equal to the reliability of the home server
- In SOA, the eight scenarios are executed by the distributed services. So higher fault tolerance than the SCA-based ones.

**Other design metrics**

- Workload
  - Usage frequency of the service scenarios
  - SOA-based HNS, it’s relatively easy to perform flexible design changes reflecting the workload

**Related Work**

- BPEL4WS
  - Standard service orchestration framework
  - Creating new service integration with XML-based language definition
  - Needs a centralized server for orchestration
  - Takes over the same problems of conventional HNS
Discussion: Advantage
- Interoperability is improved
- Fault-tolerant and load-balanced
- Easily modifying and updating the integrated service

Discussion: Limitation
- Cost of Appliances
- Communication Overhead
- Global management

Future plan
- Detect and resolve feature interaction problem
  - FI problem occurs, since multiple users can activate multiple services simultaneously in the HNS.